Minimal Local Simulation Foundations for LLM- and VLM-Driven Agents in 2D and 3D Environments

Ryuki Hyodo [1,2,3,4]

[1]SpaceData Inc., 1-17-1 Toranomon, Minato-ku, Tokyo 105-6490, Japan
[2]Graduate School of Artificial Intelligence and Science, Rikkyo University, 3-34-1 Nishi-Ikebukuro, Toshima-ku, Tokyo 171-8501, Japan
[3]Earth-Life Science Institute, Institute of Science Tokyo, 2-12-1 Ookayama, Meguro-ku, Tokyo, 152-8550, Japan
[4]Université Paris Cité, Institut de Physique du Globe de Paris, CNRS, F-75005 Paris, France
Email: ryuki.hyodo@spacedata.co.jp; hyodo@elsi.jp

Abstract

Large language models (LLMs) and vision-language models (VLMs) are expanding the range of behaviors that can be represented in agent-based simulations, but many contemporary platforms are difficult to study, modify, or run on ordinary computers. We present two intentionally minimal simulation foundations for education and rapid prototyping. *SD-AgentFoundry-2D* provides a two-dimensional multi-agent environment in which locally hosted LLM agents move, communicate, respond to place occupancy, and encounter spatially localized fire events. *SD-AgentFoundry-3D* provides a three-dimensional digital-twin environment in which a locally hosted VLM receives first-person images and produces natural-language movement instructions. Both codebases are designed to run locally on macOS, Windows, and Linux and are deliberately left open to modification rather than developed as finished applications. Together, they offer accessible starting points for learning about generative social simulation and for building domain-specific extensions.

SD-AgentFoundry-2D: https://github.com/ryukih/SD-AgentFoundry-2D
SD-AgentFoundry-3D: https://github.com/ryukih/SD-AgentFoundry-3D

1 Introduction

Agent-based simulation has long provided a bottom-up way to study how individual decisions and interactions can produce collective patterns (Bonabeau 2002). Recent progress in foundation models has widened this approach: instead of specifying every behavior as a fixed rule, a simulator can ask an LLM to interpret a situation, use natural-language memory, communicate with other agents, and choose an action (Guo et al. 2024). Social Simulacra demonstrated LLM-supported prototyping of populated social-computing systems (Park et al. 2022), while Generative Agents combined memory, reflection, and planning in an interactive two-dimensional town (Park et al. 2023). Concordia generalized language-mediated agent-based modeling through components that connect LLM calls, associative memory, and an environment-controlling game master (Vezhnevets et al. 2023). More recent systems have increased the scale and application range of this paradigm: AgentSociety and its successor provide environments for large-scale computational social experiments (Piao et al. 2025; Piao et al. 2026), and CitySim models urban schedules and collective city dynamics (Bougie & Watanabe 2025).

These developments suggest several possible uses, provided that simulated behavior is not mistaken for validated human prediction. LLM simulations of established human-subject studies

can reproduce some findings while also exhibiting systematic distortions (Aher et al. 2023). Studies that seek individual-level prediction require agents grounded in human data and evaluation against human responses (Park et al. 2024). LLM-based simulations may support exploratory studies of communication, evacuation, public policy, urban activity, and collective response to external events. They may also help organizations prototype services, examine hypothetical customer or worker interactions, and identify scenarios that deserve later evaluation with real participants. Such simulations are therefore best treated as instruments for generating hypotheses, comparing mechanisms, and rehearsing possibilities, rather than as substitutes for empirical evidence.

Textual and two-dimensional environments capture communication and social organization efficiently, but they abstract away much of the spatial and perceptual structure of physical action. VLMs and vision-language-action (VLA) models provide a complementary direction by grounding decisions in images and language. PaLM-E demonstrated an embodied multimodal language model that integrates visual, textual, and continuous state inputs for embodied reasoning (Driess et al. 2023). OpenVLA illustrates how visual and linguistic representations can be connected to actions in an open model for robot control (Kim et al. 2025). SimWorld exposes LLM and VLM agents to multimodal physical and social environments (Ren et al. 2025), CrowdVLA applies VLA agents to context-aware crowd simulation in semantically structured three-dimensional scenes (Hwang et al. 2026), and TravelAgent integrates generative agents into three-dimensional built environments for navigation and wayfinding (Noyman et al. 2024). These embodied approaches may support exploratory work on navigation, facility use, human-robot interaction, crowd behavior, and digital twins.

The growing capability of such systems also creates an educational access problem. Large platforms, remote model services, specialized simulators, and substantial compute requirements can obscure the basic loop connecting perception, reasoning, communication, action, and observation. We present *SD-AgentFoundry*, a pair of minimal local simulation foundations comprising *SD-AgentFoundry-2D* for LLM-driven multi-agent social simulation and *SD-AgentFoundry-3D* for VLM-driven embodied simulation. Our goal is not to provide another feature-complete simulator. Instead, we release two compact codebases that expose this loop directly and can serve as readable starting points. They use locally hosted models, include setup paths for major desktop operating systems, and retain deliberately simple mechanisms that users can replace. The intended contribution is an accessible foundation from which students, researchers, and practitioners can develop their own scenarios, agent designs, measurements, and interfaces.

2 System Architecture

This section describes the architecture and implementation of *SD-AgentFoundry-2D* and *SD-AgentFoundry-3D*.

2.1 SD-AgentFoundry-2D

SD-AgentFoundry-2D is a discrete two-dimensional multi-agent simulation implemented in Python (Figure 1). The environment is a bounded integer grid containing configurable places, each defined by a name, type, center, spatial extent, and capacity. The default scenario places a cafe and a library on opposite sides of the field. Agents begin outside the places at randomly generated positions and receive a simple persona field representing gender. Their available physical actions are to stay or move by one grid cell in one of four cardinal directions.

At every simulation step, each agent first identifies communicable neighbors. Communication requires both spatial proximity and a shared area: agents can communicate when they are outside all places together or when they occupy the same place, but not across place boundaries. Each agent then makes a message decision, the messages are delivered using the pre-movement neighborhood, each agent makes an action decision, and all selected movements are finally executed. Separating communication from movement gives every agent a consistent view of the current step.

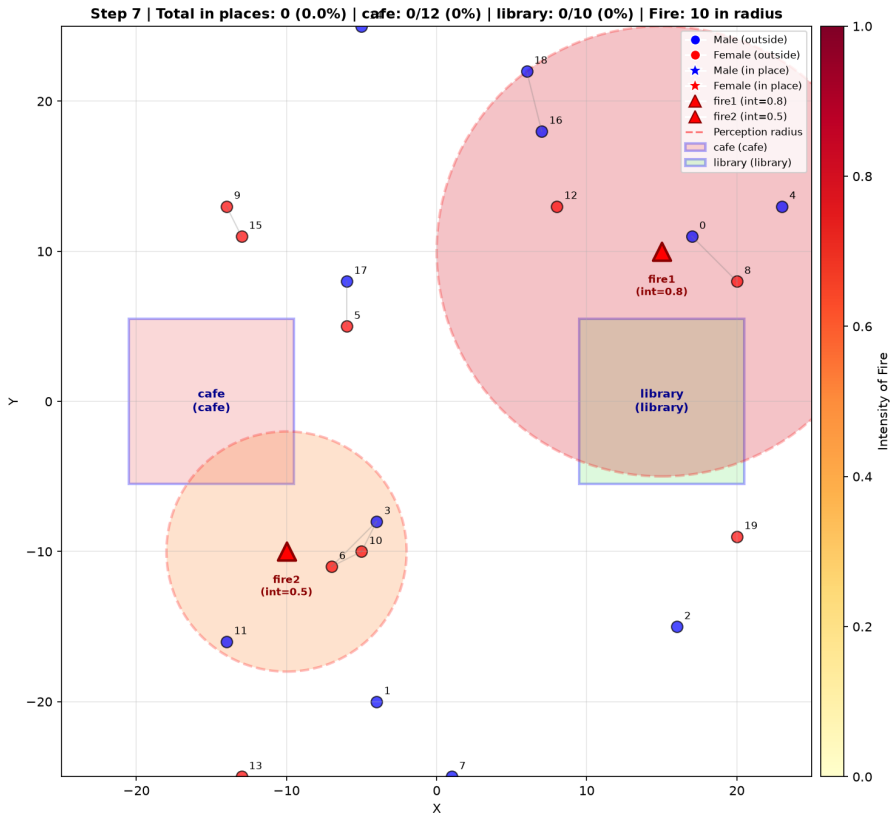

Figure 1: Snapshot of an example SD-AgentFoundry-2D simulation. The two-dimensional grid contains 20 LLM-driven agents, two places (a cafe and a library), and two active fire events. Colored markers represent the agents, shaded disks show the fire-perception radii, and gray lines connect pairs eligible for local communication. Such configurable environments may support social-simulation studies and characterization of LLM-agent behavior.

Recent memories and received messages are retained in bounded histories and included in later prompts. Such bounded histories are a deliberate simplification of a much larger design space for agent memory, which ranges from parametric internal state to retrieval-augmented external stores (Huang et al. 2026).

The prompts provide numerical state rather than qualitative prescriptions. An agent inside a place receives its population, capacity, and occupancy rate, but it is not told that the place is comfortable or crowded. Configurable fire events begin at specified steps and have positions, intensities, and perception radii. Only agents inside a fire's radius receive its numerical properties and their distance from it; agents outside the radius can learn about the event only through messages. This design leaves interpretations such as avoidance, warning, coordination, or inaction to the LLM rather than encoding them as simulator rules. Related spatial LLM-agent studies have used capacity constraints and resource hazards to examine emergent, model-dependent behavior (Takata et al. 2025; Masumori & Ikegami 2025).

The LLM is served locally through Ollama. One call determines a message and a second determines movement, memory, and reasoning, using JSON-shaped responses with conservative fallback parsing. Thus, each agent receives two distinct prompts per step: a communication prompt followed by an action-decision prompt. Detailed communication and action-decision prompts are shown in Appendix A.1. The current default uses a compact instruction-tuned Qwen model and

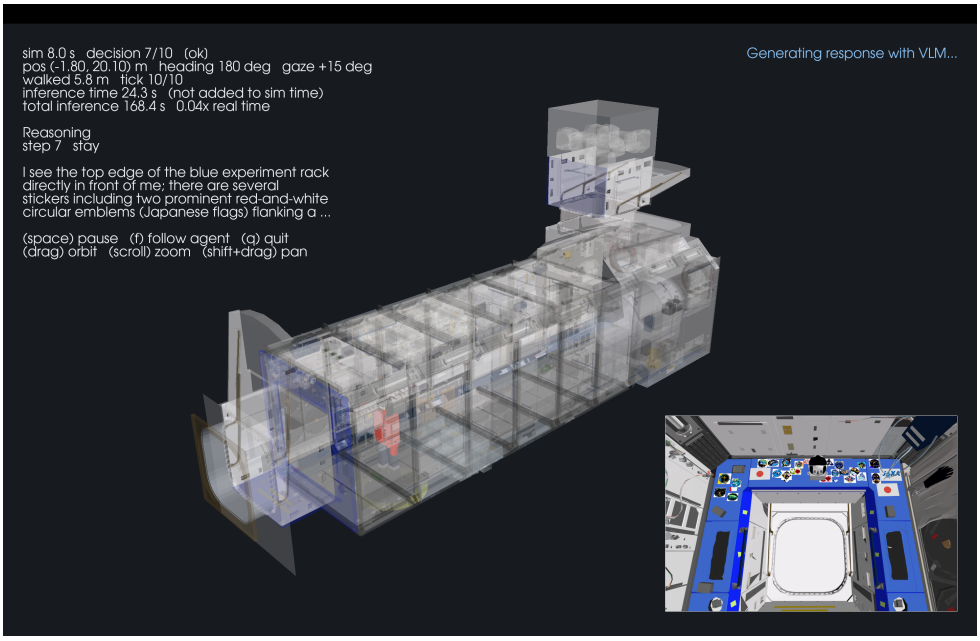

Figure 2: Snapshot of an example SD-AgentFoundry-3D simulation in the Kibo digital twin. The Kibo module is loaded as the USD environment, where a single VLM-driven agent, shown as a red-clad human avatar, attempts the assigned task. The lower-right inset is a same-viewpoint re-rendering matched to the VLM camera pose and settings. Users can replace Kibo with their own USD scenes to run VLM-driven embodied simulations in custom three-dimensional spaces.

disables model-side thinking so that the response budget is available for the requested JSON. The simulator writes memory and reasoning records to JSONL and creates a message log when communication occurs. A display-independent Matplotlib Agg renderer writes the initial state and selected post-step states as PNG files; it never opens a live window. The saved frames can be inspected directly, loaded in the separate browser viewer, or converted to video. Configuration is centralized in YAML, and setup scripts are supplied for macOS/Linux and Windows.

2.2 SD-AgentFoundry-3D

SD-AgentFoundry-3D is a three-dimensional, single-agent simulation built around Universal Scene Description (USD) assets. The default demonstration loads a digital twin of the Japanese Experiment Module "Kibo" and gives the agent the task of locating the JAXA logo and approaching it (Figure 2). The USD stage supplies geometry, scale, and coordinate information; the implementation uses `usd-core` for scene access and VTK/PyVista for rendering rather than requiring a heavyweight game engine.

At each decision point, the simulator renders the agent's first-person RGB image and sends that image and a dynamically assembled text prompt to a VLM. The text contains the task, current pose, and a bounded recent history. The detailed VLM prompt is shown in Appendix A.2. The model returns unrestricted natural language rather than a required action schema. A deterministic interpreter scans the reply and resolves each movement category from its latest occurrence in the text, since models that reason aloud state their conclusion at the end. It converts supported distance units to meters and extracts translation bearing, body rotation, gaze pitch, or a stay command. Configured limits clamp excessive motion, and unreadable responses are recorded without inventing a replacement action. This interface is a VLM-driven, VLA-style perception–decision–action loop; it is not presented as a trained VLA policy.

Table 1: Configuration and recorded artifacts for an example SD-AgentFoundry-2D run.

Setting	Value
LLM backend and model	Ollama; `qwen3:4b-instruct-2507-q4_K_M`
Inference settings	Temperature 0.2; thinking disabled; maximum 512 response tokens
Agents and inspected point	20 agents; state recorded after step 7
World	Integer grid with both axes ranging from -25 to 25
Places	Cafe at $(-15, 0)$ and library at $(15, 0)$, with capacities 12 and 10
Fire events	`fire1`: step 3, intensity 0.8, radius 15; `fire2`: step 5, intensity 0.5, radius 8
Step 7 state	Both fires active; 10 agents within at least one perception radius; 0 agents inside places
Recorded through step 7	8 PNG frames, 140 memory/reasoning records, and 88 per-recipient message-delivery records

Table 2: Configuration for an example SD-AgentFoundry-3D run.

Setting	Value
VLM backend and model	Ollama; `qwen3.6:27b`
Scene	`assets/kibou/KIBOU.usd`, excluding the distant Earth backdrop
Task	Find JAXA logo and get close to it
Initial pose	Position $(4.0, 20.1)$ m; yaw 180°; pitch 0°; eye height 1.6 m
Run length	10 VLM decisions for the representative short run
Camera	512×384 pixels; 70° vertical field of view
Output	First-person frames, raw and parsed decisions, pose history, configuration snapshot, and replay frames

The agent can translate, rotate its body, and change its gaze within configured bounds. A live view combines a semi-transparent overview of the scene, the agent and its trajectory, and an inset re-rendered from the same camera pose and settings as the first-person image supplied to the VLM. The default backend uses a local vision-capable model through Ollama, while local Transformers, OpenAI-compatible, and scripted backends provide replaceable interfaces. Inference wall-clock time is kept separate from simulation time so that model latency does not alter the simulated dynamics.

For reproducibility and auditing, every decision record preserves the raw model response, parsed command, parser notes, pre- and post-action poses, boundary effects, inference time, and the path of the input image when frame saving is enabled. The effective configuration is saved with the run. A separate replay path reconstructs overview frames from the log rather than treating the live window as a video recording. Cross-platform setup scripts are included.

3 Example Runs & Beyond

The following representative runs are intended as qualitative usage examples, not as controlled evaluations or evidence of predictive validity. Their tables document the configurations and recorded outputs. Exact repetition of the current 2D example additionally requires control of its randomly generated initial positions and personas, for which the present configuration does not expose a seed.

3.1 Representative 2D LLM Run

This subsection presents an illustrative 2D simulation using *SD-AgentFoundry-2D*. Twenty LLM agents are placed in a grid world containing two accessible places, a cafe and a library (Figure 1). Agents can converse with nearby agents by ex-

changing natural-language messages. Two fires occur during the run, and each agent uses locally available information and received messages to decide how to move and what to communicate to others. The scenario therefore provides a compact setting in which to observe how LLM agents respond to a shared hazard, choose between places, and interact with one another. Holding the scenario fixed while replacing the underlying LLM can also reveal model-dependent differences in agent behavior.

The purpose of this example is to demonstrate the simulation workflow rather than to analyze its outcome in depth. Figure 1 presents one snapshot of the run, showing the two fires and their perception radii, the spatial distribution of the 20 agents, and the pairs eligible for local communication. Table 1 summarizes the parameters and recorded outputs associated with this example. The snapshot confirms that the decision, communication, fire-perception, logging, and visualization components operate together, but it should not be interpreted as evidence of a causal behavioral pattern or as a validated model of human response because it represents only a single illustrative run rather than a dedicated behavioral study.

Related work has shown that two-dimensional or otherwise spatially explicit LLM-agent simulations can support the prototyping of populated social systems, the study of memory- and persona-conditioned behavior, and the observation of emergent interactions. Social Simulacra used simulated populations to prototype social-computing systems, while Generative Agents demonstrated socially interacting agents in a two-dimensional town, and Concordia provided a more general framework for language-mediated agent-based modeling (Park et al. 2022; Park et al. 2023; Vezhnevets et al. 2023). Spatial extensions of the El Farol Bar problem have examined collective choice under capacity constraints (Takata et al. 2025), while Sugarscape-style environments have revealed model-dependent behavior under resource scarcity and hazards (Masumori & Ikegami 2025). Larger platforms such as AgentSociety, AgentSociety 2, and CitySim further illustrate how generative agents may be used to explore collective behavior and urban dynamics (Piao et al. 2025; Piao et al. 2026; Bougie & Watanabe 2025). Accordingly, a simple two-dimensional world can serve both as a foundation for social-simulation experiments and as a controlled environment for studying the behavioral characteristics of LLM agents.

For such studies, researchers can, for example, vary persona distributions, initial layouts, place geometry and capacity, communication distance, memory length, inference settings, and the position, timing, intensity, and perception radius of hazards. More importantly, they can hold these conditions constant while changing the LLM and compare model-specific behavioral signatures, including movement preferences, action stability, sensitivity to hazards, communication frequency, warning propagation, clustering, and collective place choice. These measurements can help identify recurring tendencies or biases in how a particular LLM behaves when instantiated as an agent. They characterize behavior within the specified simulator and prompt design, however, rather than an intrinsic human-like personality or an empirically validated prediction of human populations. Rigorous comparison would additionally require controlled random seeds, repeated trials, and validation against appropriate human or observational data, as exemplified by work that evaluates human-grounded agents against participants' own responses (Park et al. 2024).

3.2 Representative 3D VLM Run

This subsection presents an illustrative 3D simulation using *SD-AgentFoundry-3D*. The Japanese Experiment Module "Kibo" is loaded as a USD scene, and a single VLM agent, represented by a human avatar in red clothing, is instructed to find the JAXA logo and approach it (Figure 2). At each decision, the agent receives a first-person RGB image together with the task prompt and recent action history. The VLM describes the scene and proposes a movement in natural language, which the simulator interprets as a predefined translation, rotation, gaze change, or stay action. The scenario therefore provides a compact example of visual grounding and embodied decision

making within a three-dimensional digital twin.

The purpose of this example is to demonstrate the simulation workflow rather than to analyze task performance in depth. Figure 2 presents one snapshot of the run: the overview shows the agent and its trajectory within the Kibo geometry, while the inset shows a same-viewpoint re-rendering matched to the VLM camera pose and settings. Table 2 summarizes the parameters and recorded outputs associated with this example. The snapshot confirms that USD-scene loading, first-person rendering, VLM inference, action interpretation, movement, logging, and visualization operate together, but it should not be interpreted as a performance benchmark or as evidence of reliable task completion because it represents only a single illustrative run rather than a dedicated evaluation study.

Related work has shown that visually grounded agents in three-dimensional environments can support research on navigation, interaction, and behavior in physical and social spaces. Habitat established a configurable, photorealistic simulation platform for embodied-AI tasks such as navigation and instruction following (Savva et al. 2019). Vision-and-language navigation grounded natural-language navigation instructions in visual observations of previously unseen environments (Anderson et al. 2018). More recent examples include SimWorld, which provides multimodal environments for autonomous agents in open-ended physical and social scenarios (Ren et al. 2025); TravelAgent, which studies navigation and wayfinding by generative agents in built environments (Noyman et al. 2024); and CrowdVLA, which addresses context-aware agent navigation and continuous locomotion in semantically structured crowd simulations (Hwang et al. 2026). EmbRACE-3K further provides a benchmark for first-person, multi-step embodied reasoning and reports substantial limitations in the zero-shot task success of contemporary VLMs (Lin et al. 2025).

These studies illustrate how three-dimensional simulation can connect language-based reasoning with spatial perception and action. Accordingly, a configurable USD world can serve both as a foundation for digital-twin experiments and as a controlled environment for studying the embodied behavioral characteristics of vision-language agents.

For such studies, researchers can, for example, replace the Kibo asset with another USD scene and vary the task instruction, initial pose, camera field of view, movement and gaze limits, recent-history length, model backend, and response interpreter. Facilities, streets, workplaces, or other digital twins could thereby be used to study visual grounding, object search, wayfinding, viewpoint dependence, and sensitivity to language or scene layout. More importantly, researchers can hold the scene, task, and action constraints constant while changing the VLM and compare model-specific behavioral signatures, including object-recognition errors, exploration strategies, movement efficiency, action consistency, recovery from incorrect decisions, and sensitivity to viewpoint. Perspective-taking is also an explicit dimension of comprehensive VLM spatial-reasoning benchmarks (Jia et al. 2026). These measurements characterize behavior within the specified simulator, prompt, and action interface rather than general visual intelligence or real-world embodied competence. Rigorous comparison would additionally require controlled initial conditions, repeated trials, task-specific success criteria, and validation in appropriate physical or observational settings.

The present VLM-driven loop should not be confused with a trained VLA policy. Here, a VLM produces natural-language text, and a deterministic interpreter maps recognized phrases onto a predefined set of translations, rotations, gaze changes, and stay actions. In contrast, VLA models such as RT-2 and OpenVLA are trained with robot trajectories or demonstrations so that visual and linguistic inputs are connected directly to embodiment-specific actions and transferable visuomotor skills (Brohan et al. 2023; Kim et al. 2025). CrowdVLA similarly introduces learned motion skills to bridge symbolic decisions and continuous locomotion in crowd simulation (Hwang et al. 2026).

Consequently, *SD-AgentFoundry-3D* is suitable for interpretable studies of perception, high-level decision making, and bounded simulated navigation, but it cannot by itself learn a new mo-

tor skill, generate joint- or end-effector-level control, handle contact-rich manipulation, or close a high-frequency sensorimotor loop. Those capabilities require a VLA or another trained low-level control policy, an embodiment-specific interface, and separate physical validation. Generative Pretrained Controllers exemplify a distinct learned low-level policy that maps reusable motion representations to physics-based character control (Shi et al. 2026).

4 Summary

LLM- and VLM-driven agents make it possible to explore social interaction and embodied decision making with mechanisms that are difficult to express as fixed behavioral rules. Such simulations may inform hypothesis generation for social challenges, urban and facility planning, emergency communication, service prototyping, and other business scenarios, but their outputs require empirical validation before they can support real-world claims. The two *SD-AgentFoundry* foundations expose complementary levels of abstraction: *SD-AgentFoundry-2D* emphasizes communication and collective behavior in a transparent two-dimensional world, whereas *SD-AgentFoundry-3D* emphasizes visual grounding and physical action in a three-dimensional digital twin.

Both systems are intentionally basic, locally operated, and designed for use on macOS, Windows, and Linux. Their purpose is educational accessibility and extensibility rather than completeness. By publishing readable implementations of the perception–reasoning–action loop, configurable scenarios, visualizations, and auditable logs, we hope to lower the barrier to studying these techniques and to encourage users to add their own environments, agent architectures, models, measurements, and application-specific safeguards.

Acknowledgment

R.H. acknowledges financial support from JSPS KAKENHI Grant Numbers 26K00756, 23KK0253, 22K14091, 21H04512, 21H04514, and 20KK0080. ChatGPT was used to assist with English-language proofreading of the manuscript.

References

Aher, Gati V., Rosa I. Arriaga & Adam Tauman Kalai (2023). "Using Large Language Models to Simulate Multiple Humans and Replicate Human Subject Studies". In: *Proceedings of the 40th International Conference on Machine Learning*. Vol. 202. Proceedings of Machine Learning Research. PMLR, pp. 337–371. URL: https://proceedings.mlr.press/v202/aher23a.html.

Anderson, Peter et al. (2018). "Vision-and-Language Navigation: Interpreting Visually-Grounded Navigation Instructions in Real Environments". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3674–3683. DOI: 10.1109/CVPR.2018.00387. URL: https://openaccess.thecvf.com/content_cvpr_2018/html/Anderson_Vision-and-Language_Navigation_Interpreting_CVPR_2018_paper.html.

Bonabeau, Eric (2002). "Agent-Based Modeling: Methods and Techniques for Simulating Human Systems". In: *Proceedings of the National Academy of Sciences* 99 (suppl. 3), pp. 7280–7287. DOI: 10.1073/pnas.082080899. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC128598/.

Bougie, Nicolas & Narimasa Watanabe (2025). "CitySim: Modeling Urban Behaviors and City Dynamics with Large-Scale LLM-Driven Agent Simulation". In: *arXiv preprint arXiv:2506.21805*. DOI: 10.48550/arXiv.2506.21805. arXiv: 2506.21805.

Brohan, Anthony et al. (2023). "RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control". In: *arXiv preprint arXiv:2307.15818*. DOI: 10.48550/arXiv.2307.15818. arXiv: 2307.15818.

Driess, Danny et al. (2023). "PaLM-E: An Embodied Multimodal Language Model". In: *Proceedings of the 40th International Conference on Machine Learning*. Vol. 202. Proceedings of Machine Learning Research. PMLR, pp. 8469–8488. URL: https://proceedings.mlr.press/v202/driess23a.html.

Guo, Taicheng et al. (2024). "Large Language Model based Multi-Agents: A Survey of Progress and Challenges". In: *Proceedings of IJCAI-24*, pp. 8048–8057. DOI: 10.24963/ijcai.2024/890. arXiv: 2402.01680.

Huang, Wei-Chieh et al. (2026). "A Survey of Agent Memory in the Second Half: Towards Self-Evolving and Long-Horizon Agents". In: *Transactions on Machine Learning Research*. DOI: 10.48550/arXiv.2602.06052. arXiv: 2602.06052.

Hwang, Juyeong et al. (2026). "CrowdVLA: Embodied Vision-Language-Action Agents for Context-Aware Crowd Simulation". In: *arXiv preprint arXiv:2604.05525*. DOI: 10.48550/arXiv.2604.05525. arXiv: 2604.05525.

Jia, Mengdi et al. (2026). "OmniSpatial: Towards Comprehensive Spatial Reasoning Benchmark for Vision Language Models". In: *The Fourteenth International Conference on Learning Representations*. DOI: 10.48550/arXiv.2506.03135. arXiv: 2506.03135.

Kim, Moo Jin et al. (2025). "OpenVLA: An Open-Source Vision-Language-Action Model". In: *Proceedings of The 8th Conference on Robot Learning*. Vol. 270. Proceedings of Machine Learning Research. PMLR, pp. 2679–2713. URL: https://proceedings.mlr.press/v270/kim25c.html.

Lin, Mingxian et al. (2025). "EmbRACE-3K: Embodied Reasoning and Action in Complex Environments". In: *arXiv preprint arXiv:2507.10548*. DOI: 10.48550/arXiv.2507.10548. arXiv: 2507.10548.

Masumori, Atsushi & Takashi Ikegami (2025). "Do Large Language Model Agents Exhibit a Survival Instinct? An Empirical Study in a Sugarscape-Style Simulation". In: *arXiv preprint arXiv:2508.12920*. DOI: 10.48550/arXiv.2508.12920. arXiv: 2508.12920.

Noyman, Ariel, Kai Hu & Kent Larson (2024). "TravelAgent: Generative Agents in the Built Environment". In: *arXiv preprint arXiv:2412.18985*. DOI: 10.48550/arXiv.2412.18985. arXiv: 2412.18985.

Park, Joon Sung et al. (2022). "Social Simulacra: Creating Populated Prototypes for Social Computing Systems". In: *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–18. DOI: 10.1145/3526113.3545616. arXiv: 2208.04024.

Park, Joon Sung et al. (2023). "Generative Agents: Interactive Simulacra of Human Behavior". In: *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–22. DOI: 10.1145/3586183.3606763. arXiv: 2304.03442.

Park, Joon Sung et al. (2024). "LLM Agents Grounded in Self-Reports Enable General-Purpose Simulation of Individuals". In: *arXiv preprint arXiv:2411.10109*. DOI: 10.48550/arXiv.2411.10109. arXiv: 2411.10109.

Piao, Jinghua et al. (2025). "AgentSociety: Large-Scale Simulation of LLM-Driven Generative Agents Advances Understanding of Human Behaviors and Society". In: *arXiv preprint arXiv:2502.08691*. DOI: 10.48550/arXiv.2502.08691. arXiv: 2502.08691.

Piao, Jinghua et al. (2026). "AgentSociety 2: An Integrated Research Environment for Executable Social Science". In: *arXiv preprint arXiv:2607.11895*. DOI: 10.48550/arXiv.2607.11895. arXiv: 2607.11895.

Ren, Jiawei et al. (2025). "SimWorld: An Open-Ended Realistic Simulator for Autonomous Agents in Physical and Social Worlds". In: *arXiv preprint arXiv:2512.01078*. DOI: 10.48550/arXiv.2512.01078. arXiv: 2512.01078.

Savva, Manolis et al. (2019). "Habitat: A Platform for Embodied AI Research". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9339–9347. DOI: 10.1109/ICCV.2019.00943. URL: https://openaccess.thecvf.com/content_ICCV_2019/html/Savva_Habitat_A_Platform_for_Embodied_AI_Research_ICCV_2019_paper.html.

Shi, Yi et al. (2026). "GPC: Large-Scale Generative Pretraining for Transferable Motor Control". In: *SIGGRAPH Conference Papers '26*. DOI: 10.1145/3799902.3811038. arXiv: 2606.29148.

Takata, Ryosuke, Atsushi Masumori & Takashi Ikegami (2025). "Emergent Social Dynamics of LLM Agents in the El Farol Bar Problem". In: *arXiv preprint arXiv:2509.04537*. DOI: 10.48550/arXiv.2509.04537. arXiv: 2509.04537.

Vezhnevets, Alexander Sasha et al. (2023). "Generative Agent-Based Modeling with Actions Grounded in Physical, Social, or Digital Space Using Concordia". In: *arXiv preprint arXiv:2312.03664*. DOI: 10.48550/arXiv.2312.03664. arXiv: 2312.03664.

A Prompts Used in the Simulations

This appendix documents the text supplied to the local models. Angle-bracketed expressions (e.g., <xxx>) in the templates denote values inserted at runtime. Blocks explicitly marked as conditional are omitted when the corresponding information is unavailable. Line breaks inside the framed listings are part of the prompts, whereas automatic visual wrapping does not insert additional characters.

A.1 SD-AgentFoundry-2D LLM Prompts

Each agent makes two sequential LLM calls during a simulation step. The first, a communication prompt (Section A.1.1), determines what the agent says to locally communicable agents and deliberately excludes their coordinates. The selected messages are delivered before movement. The second, an action-decision prompt (Section A.1.2), includes spatial coordinates and the exchanged messages and requests a movement or stay action, memory, and reasoning. These two sections reproduce the fixed text in the implementation and identify every dynamic or conditional field. Section A.1.3 then provides a reconstructed example showing how representative runtime values populate the action-decision template.

A.1.1 Communication prompt template

```
You are Agent <agent-id> (<gender>) in a 2D world with multiple places (<comma-separated unique
    place types>).

=== YOUR CURRENT STATE ===
Gender: <gender>
In place: <Yes or No>
<Current place: place-name; omitted when outside>

<CONDITIONAL WHEN INSIDE A PLACE>
You are currently in the <place-type> (<place-name>).
  Number of agents here: <agents-in-place>
  Capacity: <capacity>
  Occupancy rate: <occupancy-rate rounded to two decimals>

<CONDITIONAL WHEN ONE OR MORE FIRES ARE PERCEIVED>
=== FIRE EVENT ===
<REPEATED FOR EACH PERCEIVED FIRE>
Fire "<fire-name>":
  Position: (<fire-x>, <fire-y>)
  Intensity: <intensity> (scale: 0.0 to 1.0)
  Radius: <radius>
  Your distance: <distance rounded to two decimals>

=== NEARBY AGENTS (you can communicate with these agents) ===
<Agent id (gender) is in place-name (place-type), Agent id (gender) is outside the places, or No
    nearby agents.>

=== PREVIOUS MEMORY ===
<- one line per retained memory, or No previous experiences.>

=== MESSAGES FROM OTHERS ===
<from Agent id: message, or No messages received.>
```

```
=== YOUR TASK ===
Decide what message you want to send to nearby agents. You can share your observations, experiences
    , or thoughts about the places and situation.

=== RESPOND IN JSON ===
{
    "message": "message to nearby agents (max 200 words, optional if you don't want to send a
    message)",
    "reasoning": "brief explanation of why you want to send this message"
}

Step: <step>
```

A.1.2 Action-decision prompt template

```
You are Agent <agent-id> (<gender>) in a 2D world with multiple places (<comma-separated unique
    place types>).

=== YOUR CURRENT STATE ===
Gender: <gender>
Position: (<x>, <y>)
In place: <Yes or No>
<Current place: place-name; omitted when outside>

<CONDITIONAL WHEN INSIDE A PLACE>
You are currently in the <place-type> (<place-name>).
  Number of agents here: <agents-in-place>
  Capacity: <capacity>
  Occupancy rate: <occupancy-rate rounded to two decimals>

<CONDITIONAL WHEN ONE OR MORE FIRES ARE PERCEIVED>
=== FIRE EVENT ===
<REPEATED FOR EACH PERCEIVED FIRE>
Fire "<fire-name>":
  Position: (<fire-x>, <fire-y>)
  Intensity: <intensity> (scale: 0.0 to 1.0)
  Radius: <radius>
  Your distance: <distance rounded to two decimals>

=== PLACE LOCATIONS ===
<place-name> (<place-type>): center at (<center-x>, <center-y>), covers X from <minimum-x> to <
    maximum-x>, Y from <minimum-y> to <maximum-y>
<one line for every configured place>

=== NEARBY AGENTS ===
<Agent id (gender) is at (x, y) and is in place-name (place-type), Agent id (gender) is at (x, y)
    and is outside the places, or No nearby agents.>

=== PREVIOUS MEMORY ===
<- one line per retained memory, or No previous experiences.>

=== MESSAGES FROM OTHERS ===
<from Agent id: message, or No messages received.>

<CONDITIONAL WHEN A MESSAGE WAS CHOSEN>
=== MESSAGE YOU DECIDED TO SEND ===
```

```
<message selected by the communication call>
=== AVAILABLE ACTIONS ===
- "stay": remain at current position
- "move" with direction: "up" (Y+1), "down" (Y-1), "left" (X-1), "right" (X+1)

Field boundaries: X and Y from -<half-space-size> to +<half-space-size>

=== RESPOND IN JSON ===
{
    "action": "move" or "stay",
    "direction": "up", "down", "left", or "right" (only if action is "move"),
    "memory": "what you want to remember for the next step (your thoughts, observations, intentions
    )",
    "reasoning": "brief explanation of your decision"
}

Step: <step>
```

A.1.3 Example action prompt

The 2D implementation does not retain the exact prompts sent to Ollama. The following is therefore not a historical record or an observed result. It is a representative prompt generated with the inspected code and current cafe/library place definitions at step 7, after `fire1` has activated. Female Agent 3 is at $(14, 2)$ in the library with two other occupants, one retained memory, one received message, nearby Agent 1, and perceived fire event `fire1` at a distance of 8.06 grid units. The order in which distinct place types appear in the opening sentence can vary because the implementation derives that list through set iteration.

```
You are Agent 3 (female) in a 2D world with multiple places (cafe, library).

=== YOUR CURRENT STATE ===
Gender: female
Position: (14, 2)
In place: Yes
Current place: library

You are currently in the library (library).
  Number of agents here: 3
  Capacity: 10
  Occupancy rate: 0.30

=== FIRE EVENT ===
Fire "fire1":
  Position: (15, 10)
  Intensity: 0.8 (scale: 0.0 to 1.0)
  Radius: 15
  Your distance: 8.06

=== PLACE LOCATIONS ===
cafe (cafe): center at (-15, 0), covers X from -20 to -10, Y from -5 to 5
library (library): center at (15, 0), covers X from 10 to 20, Y from -5 to 5

=== NEARBY AGENTS ===
Agent 1 (male) is at (16, 1) and is in library (library)

=== PREVIOUS MEMORY ===
```

```
- I entered the library to inspect its occupancy.

=== MESSAGES FROM OTHERS ===
from Agent 1: I can also see fire1 from inside the library.

=== MESSAGE YOU DECIDED TO SEND ===
I can detect fire1 near the library; please stay alert.
=== AVAILABLE ACTIONS ===
- "stay": remain at current position
- "move" with direction: "up" (Y+1), "down" (Y-1), "left" (X-1), "right" (X+1)

Field boundaries: X and Y from -25 to +25

=== RESPOND IN JSON ===
{
    "action": "move" or "stay",
    "direction": "up", "down", "left", or "right" (only if action is "move"),
    "memory": "what you want to remember for the next step (your thoughts, observations, intentions
    )",
    "reasoning": "brief explanation of your decision"
}

Step: 7
```

A.2 SD-AgentFoundry-3D VLM Prompt

The VLM receives a dynamically assembled text prompt together with the current first-person RGB image. Section A.2.1 presents the text-prompt template. Its recent-history block is limited by the configured memory size; each entry contains the earlier raw reply flattened to one line and truncated to at most 160 characters, the interpreter's outcome, and the resulting pose. The final nudge is included only when the latest three retained commands are all idle. Section A.2.2 provides an example assembled for decision step 0 of the current Kibo configuration, when no action history is yet available.

A.2.1 Dynamic text-prompt template

```
<task prompt from config.yaml>

Movement conventions: distances are in metres unless you name another unit; angles are in degrees.
    Turning left is counter-clockwise. You can look up or down, but not straight up or straight
    down.

Your current pose: position (<x>, <y>) m, heading <yaw> deg, gaze <signed-pitch> deg.

<CONDITIONAL WHEN RECENT HISTORY EXISTS>
What you did recently:
- step <decision-index>: you said "<flattened reply>" -> <interpreted outcome>. You were then at <
    pose summary>.
<one line per retained history entry>

<CONDITIONAL AFTER THREE IDLE DECISIONS>
You have not moved for several steps. Try something different.

What is your next single movement?
```

A.2.2 Example text prompt

The following is the text assembled for decision step 0 when the current task prompt in `config.yaml` is combined with the logged initial pose $(x, y, z) = (4.0, 20.1, -0.2358)$, yaw 180°, and pitch 0°. The prompt exposes only the two-dimensional foot position, heading, and gaze. Because no earlier decision exists at step 0, there is no history block. The VLM receives this text together with the corresponding first-person RGB image.

```
You are an agent standing inside the Japanese Experiment Module "Kibo" of the ISS.
The image is your own first-person view.
Task: Find JAXA logo and get close to it.
Briefly say what you see and why you are moving, then state your next single
movement in one sentence, LAST.
Examples of the movement sentence: "move forward 2 m", "turn right 30 degrees",
"look up 30 degrees", "stay".

Movement conventions: distances are in metres unless you name another unit; angles are in degrees.
    Turning left is counter-clockwise. You can look up or down, but not straight up or straight
    down.

Your current pose: position (4.00, 20.10) m, heading 180 deg, gaze +0 deg.

What is your next single movement?
```